Multilingual Weighted Codebooks for Non-native Speech Recognition

Martin Raab[1,2], Rainer Gruhn[1,3], Elmar Noeth[2]

[1] Harman Becker Automotive Systems, Speech Dialog Systems, Ulm, Germany
mraab@harmanbecker.com,
http://www.harmanbecker.de
[2] University of Erlangen, Dept. of Pattern Recognition, Erlangen, Germany
[3] University of Ulm, Dept. of Information Technology, Ulm, Germany

Abstract. In many embedded systems commands and other words in the user's main language must be recognized with maximum accuracy, but it should be possible to use foreign names as they frequently occur in music titles or city names. Example systems with constrained resources are navigation systems, mobile phones and MP3 players.
Speech recognizers on embedded systems are typically semi-continuous speech recognizers based on vector quantization. Recently we introduced Multilingual Weighted Codebooks (MWCs) for such systems. Our previous work shows significant improvements for the recognition of multiple native languages. However, open questions remained regarding the performance on non-native speech.
We evaluate on four different non-native accents of English, and our MWCs produce always significantly better results than a native English codebook. Our best result is a 4.4% absolute word accuracy improvement. Further experiments with non-native accented speech give interesting insights in the attributes of non-native speech in general.

Key words: multilingual, codebook, semi-continuous, non-native

1 Introduction

For speech recognition in embedded systems, multilinguality is a great challenge. Commands and other words in the user's main language must be recognized with maximum possible accuracy, but also words in other languages should be recognized well, e.g. for music titles. Therefore a system is needed that performs as well as possible for all languages under the constraint of keeping monolingual performance in the main language. Earlier approaches [1–3] did not address the constraint of conserving the main language performance when traversing from a mono- to a multi-lingual system.

An additional problem is that human users are uttering names in foreign languages. In most cases such pronunciations will differ significantly from native pronunciations of the same name. These deviations in non-native speech are well known to degrade the performance of speech recognizers severely.

We showed that MWCs can outperform traditional codebook generation methods for native speech in [4]. Results in this paper show that a similar conclusion can be drawn for non-native speech. In addition, our results show that it is better to use more languages than the spoken language and the native language of the speaker for an MWC for non-native speech.

For completeness, the motivation and the idea of MWCs is briefly recapitulated. The motivation is based on the fact that traditional vector quantization algorithms like the LBG [5] are not optimal for our scenario. The aim of the LBG is to find a limited number of Gaussian prototypes in the feature space that cover the training data as well as possible. For the multilingual scenario, either only main language training data or data from all languages can be used with the LBG. In the first case the codebook is only optimized for the main language, not considering the performance on the additional languages. In the second case the codebook is optimized for all languages without prioritizing the main language.

Therefore the idea of MWCs is to generate a codebook that optimizes performance on all languages, with the constraint that the main language performance is more important than the performance on other languages. The first step is the construction of a codebook for each language. For this soft vector quantization based on the LBG approach is used. From these initial codebooks a new codebook is created. As this new codebook is based on codebooks from many languages, it is called multilingual, and as the influence of each original codebook can be adjusted, it is called weighted.

The remainder of this paper is organized as follows: Section 2 describes the baseline architecture to train recognizers for multiple languages. Section 3 explains how MWCs are constructed from initial codebooks. Section 4 describes our experimental setup. The results are given in Section 5. Finally, a conclusion is drawn.

2 Baseline System

We start with a well trained monolingual semi-continuous HMM speech recognizer. While keeping the codebook which was generated with the main language constant, the following is done for each additional language:

- Add all additional language HMMs to the recognizer
- Train these additional HMMs with training data from the corresponding language, not changing the codebook

Finally, we have a system with trained HMMs for all languages. As we explained in the introduction the main language codebook is not optimal for our scenario.

3 Extended System

To improve the performance on the additional languages, the monolingual codebook is replaced by a MWC. The MWC is basically the main language codebook plus some additional Gaussians. Figure 1 depicts an example for the extension of a codebook to cover an additional language. From left to right one iteration of the generation of MWCs is represented.

The picture to the left shows the initial situation. The Xs are mean vectors from the main language codebook, and the area that is roughly covered by them is indicated by the dotted line. Additionally, the numbered Os are mean vectors from the second language codebook. Supposing that both Xs and Os are optimal for the language they were created for, it is clear that the second language contains sound patterns that are not typical for the first language (Os 1,2 and 3).

The middle picture shows the distance calculation. For each of the second language codebook vectors, the nearest neighbor among the main language Gaussians is determined. These nearest neighbor connections are indicated by the dotted lines. Our previous experiments showed that using the Mahalanobis distance produces the best results [4].

The right picture presents the outcome of one iteration. From each of the nearest neighbor connections, the largest one (O number 2) was chosen as this is obviously the mean vector which causes the largest vector quantization error. Thus, the Gaussian O number 2 was added to the main language codebook.

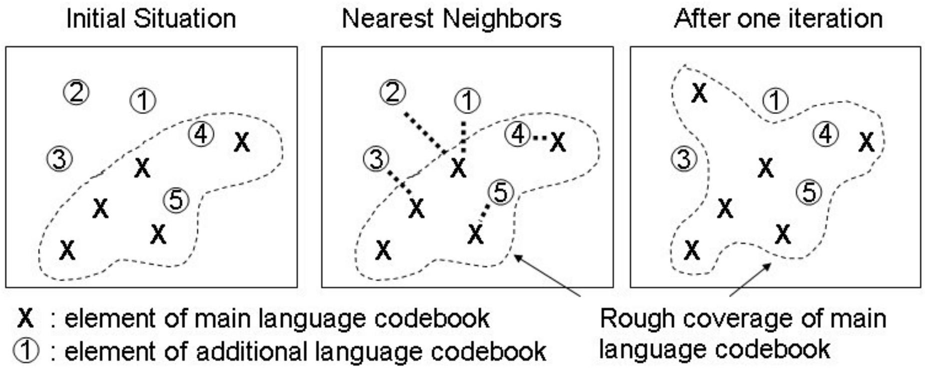
Initial Situation
Nearest Neighbors
After one iteration
①
②
③
④
⑤
X
X : element of main language codebook
① : element of additional language codebook
Rough coverage of main language codebook

Fig. 1. Basic Idea of Multilingual Weighted Codebooks

4 Experimental Setup

Our semi-continuous speech recognizer uses 11 MFCCs with their first and second derivatives per frame and LDA for feature space transformation. All recognizers are trained on 200 hours of Speecon data [6]. The HMMs are context dependent.

We used five languages for the training of our recognizer (US English, French, German, Spanish and Italian). For each training language a codebook with 1024

Gaussians was created by the LBG algorithm. In Section 5.1 some results on native speech are given. The native test sets consist of city names. The noise conditions are uniform within each language, but not across languages.

For the non-native tests in Section 5.2 the HIWIRE data [7] is used. We test on the clean speech test which is provided with the data (50% of the HIWIRE data). Our results are lower than other published results, which we believe is due to the fact that we trained on Speecon, which contains some background noise. On some tests on low noise HIWIRE speech our systems clearly outperformed results in the literature. The HIWIRE database contains English from French, Spanish, Italian and Greek speakers. The adaptation data is not used.

To reduce the number of experiments, only results with German as the main user interaction language are presented.

5 Results

In this results section we want to give a complete overview of the benefits of MWCs for multilingual embedded systems. Therefore, both results of MWC systems on native and on non-native speech are presented. In addition, we present some further results that are necessary to answer some questions raised by the results on native and non-native speech in a plausible manner.

5.1 MWCs on Native Speech

The performance is evaluated on German, English, Italian, French and Spanish test sets. German is the main language for the MWC construction. The MWC algorithm can only take two codebooks as input. Therefore we put all Gaussians from the additional languages in a large codebook with 4096 Gaussians. Together with the German codebook this is the input to the MWC algorithm.

Table 1 shows the results of the baseline and several MWC systems. The baseline experiment uses the 1024 German Gaussians as codebook. The other systems add 200, 400 and 800 Gaussians from the additional languages. Thus, the total codebook sizes are 1224, 1424 and 1824. The first column with word accuracies shows that the performance on the German test set varies insignificantly. This is what we expected, as the LBG produces already an optimal

Table 1. Word Accuracies with MWCs on English

Total Gaussians	Added Gaussians	German	English	Italian	French	Spanish
1024	0	84.1	65.6	85.2	68.7	88.3
1224	200	83.8	68.4	88.3	69.0	90.2
1424	400	84.0	70.9	87.9	71.3	**91.5**
1824	800	**84.3**	**72.0**	**89.7**	**72.9**	91.0

codebook for German. Thus, the extensions to the codebook can not improve performance on German, but they also do not hurt.

However, for the additional languages significant improvements are achieved. In general, the performance increase is correlated to the amount of Gaussians we add to the codebook, meaning the more we add, the better the performance. The differences between the different test sets are less relevant, as they are due to different noise conditions and on different languages.

It is also important to verify how the MWC algorithm performs compared to a common codebook that is trained with data from all languages. Therefore we build a codebook with 1424 Gaussians with the LBG algorithm. The results of this system are presented in Table 2, the baseline is the 1424 MWC system. For the additional languages, the codebook created by the LBG outperforms our new algorithm if performance over all four languages is compared. However, the performance on the additional languages was only our second priority aim. On German, the main language, our MWC significantly outperforms the LBG approach. Thus the MWC systems are better for multilingual systems that have one main interaction language.

Table 2. Comparison to multilingual Codebook created only with LBG

Codebook	German	English	Italian	French	Spanish
1424 MWC	**84.0**	**70.9**	87.9	71.3	**91.5**
1424 LBG	80.8	70.5	**90.6**	**72.2**	91.4

5.2 MWCs on Non-native Speech

Table 3 shows the performance on non-native English with Spanish, French, Italian and Greek accent. The first column shows for reference the performance on native English. Again, German is the main language for the multilingual systems. Thus there are two systems that can be regarded as baseline. First, the system with an English codebook, as we want to improve performance compared to a system that would usually be used to recognize non-native English. On the other hand, a system with only the German codebook can also be regarded as baseline, as all our MWC systems base on the German codebook. The MWC systems are the same as in Section 5.1.

A completely unanticipated result is the performance of the two baseline systems on non-native speech. While for native English a native English codebook is significantly better, the system using the German codebook is in all cases better for the recognition of non-native English. This shows how strong non-native speech differs from native English.

But we also have to analyze the results for each of our baselines separately. When the focus is on the performance relative to the system with the English

Table 3. Word Accuracies with MWCs on HIWIRE

Codebook	US City	Hiwire SP	Hiwire FR	Hiwire IT	Hiwire GR
English 1024	**75.5**	82.5	83.9	81.6	83.1
German 1024	65.6	85.4	**86.9**	82.8	85.5
5ling_1224	68.6	86.2	86.6	84.6	85.3
5ling_1424	68.4	86.4	86.7	**85.7**	85.8
5ling_1824	70.9	**86.9**	86.2	84.2	**86.3**

codebook, the MWCs steadily improve the performance. The larger the codebook is, the better the performance. The MWC system with 1824 Gaussians shows significant improvements for all accents. Thus we have shown that MWCs are a very valuable tool for the recognition of non-native speech.

However, when the focus is on the performance relative to a system with German codebook, the improvements on the non-native accents are rather moderate. To verify all the results we also evaluate each test set in a phoneme loop and align the recognition phonemes via dynamic programming to the reference phoneme string. As the HIWIRE corpus is not phonetically transcribed, the canonical phoneme sequence of each word in the reference is used as the reference phoneme string.

Table 4. Phoneme Loop Accuracies on HIWIRE

Codebook	Hiwire SP	Hiwire FR	Hiwire IT	Hiwire GR
English 1024	-1.1	8.1	-0.9	-6.4
German 1024	3.8	10.5	3.3	0.1
5ling_1224	5.9	12.8	5.1	1.0
5ling_1424	5.5	12.9	5.0	0.6
5ling_1824	**6.6**	**13.1**	**6.1**	**1.8**

The results in Table 4 confirm the previous results, and the effect that the German codebook produces better performance than an English codebook is even more articulated. Obviously, all phoneme accuracies are in a very low range due to high insertion rates (the phone correctness rates for the classifiers are about 50%). These high insertion rates are due to two reasons. First, phoneme recognition on non-native speech is just a difficult problem. Second, we usually apply phonotactical models to keep insertion rates lower. However, for the experiments in Table 4, we did not want to influence the recognition rates by additional rules.

In the next section we analyze why a German codebook performs better than an English codebook and present a plausible explanation.

5.3 Native language codebooks on Non-native Speech

This set of experiments shows results of systems with codebooks that are built purely on native data of each language, each with 1024 Gaussians. The HMMs are trained with native English speech. The results show that for the native US City test, the US codebook is by far the best. Yet for non-native speech, the US English codebook performs worse than other codebooks. The strongest contrast can be observed for the Spanish case. The Spanish codebook loses almost 20% word accuracy on the native English test. On the other hand, the Spanish codebook performs better than the English codebook for English with Spanish accent.

Table 5. Word Accuracies with native language codebooks

Codebook	US City	Hiwire SP	Hiwire FR	Hiwire IT	Hiwire GR
German	65.6	85.4	**86.9**	82.8	85.5
Italian	62.8	**87.1**	84.3	**86.2**	**86.5**
Spanish	56.1	85.5	79.6	80.0	82.9
French	64.2	86.7	86.0	83.0	84.6
US English	**75.5**	82.5	83.9	81.6	83.1

These results show how important it is to have a good codebook, and how strong non-native English differs from native English. The German and Italian codebook outperform the English codebook significantly in all cases. A possible explanation is that a codebook performs better the more phonemes the corresponding language has. In our phoneme sets, which are based on standard SAMPA [8] German has 59 Phonemes, Italian 50, English 46, French 37 and Spanish 29.

Apart from this unexpected aspect of non-native speech, the results also provide evidence for the common notion that non-native speakers use sounds of their mother tongue. The codebook built on the mother tongue of the speaker always performs very good, even if the language itself has fewer phonemes. The Spanish codebook system achieves 85.5% on the Spanish accented English, and the Italian codebook system achieves by far the best performance on the Italian accented speech.

6 Conclusion

This paper evaluates MWCs, a technique for improved vector quantization that has recently been introduced for multilingual scenarios. We showed that an MWC increases performance significantly on native speech of several languages while keeping monolingual performance for the main language. Furthermore, we were able to show that MWCs show significantly better performance on non-native English than a native English codebook.

The fact that a German codebook performed better than an English codebook on non-native English was a surprising aspect of our results. In Section 5.3 we showed that the performance of a codebook on non-native English is related to the amount of phonemes of the language that is used for building the codebook. In other words, building codebooks on phonetically rich languages is better for non-native speech recognition.

This gives an interesting insight into the attributes of non-native speech, being of interest for all work on non-native speech. It seems, that the sounds produced by non-native speakers are not only from their native language or the language they want to speak. Additionally, when they fail to produce foreign sounds they produce sounds that are typical for humans in general and easy to articulate.

We are aware that we have only evaluated several accents of English, but if we trust the conclusion above, we can generalize our results further. As MWCs can cover typical human sounds of all languages, not limited to the phonetical richness of one language, the conclusion would be that the MWC algorithm can produce optimal codebooks for non-native speech of all languages.

7 Acknowledgments

We thank the HIWIRE project from the EC 6th Framework IST Programme for the non-native test data [7].

References

1. Koehler, J.: Multilingual phone models for vocabulary-independent speech recognition tasks. Speech Communication Journal **35**(1-2) (2001) 21–30
2. Gruhn, R., Markov, K., Nakamura, S.: A statistical lexicon for non-native speech recognition. In: Proc. Interspeech, Jeju Island, Korea (2004) 1497–1500
3. Schultz, T., Waibel, A.: Language-independent and language-adaptive acoustic modeling for speech recognition. Speech Communication **35** (2001) 31–51
4. Raab, M., Gruhn, R., Noeth, E.: Multilingual weighted codebooks. In: Proc. ICASSP, Las Vegas, USA (2008)
5. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantization design. IEEE Transactions on Communications **28**(1) (1980) 84–95
6. Iskra, D., Grosskopf, B., Marasek, K., van den Huevel, H., Diehl, F., Kiessling, A.: Speecon - speech databases for consumer devices: Database specification and validation. In: Proc. LREC. (2002)
7. Segura, J., et al.: The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication (2007) `http://www.hiwire.org/`.
8. Wells, J.: SAMPA (2008) `http://www.phon.ucl.ac.uk/home/sampa/index.html`.